

Aplicația II.4

INSTRUMENTE VIRTUALE PENTRU GENERAREA ȘI VIZUALIZAREA SEMNALELOR ȘI A GRAFICELOR DE FUNCTII MATEMATICE

4.1. Semnale; tipuri și caracteristici

Generarea de semnale cu anumite caracteristici predefinite prezintă o importanță deosebită pentru numeroase sisteme de măsurare și testare. Tipurile uzuale de semnale utilizate în astfel de aplicații sunt corespunzătoare următoarelor forme de undă: sinusoidală, rectangulară, triunghiulară, în dinți de fierăstrău, rampă și impuls. În anumite cazuri pot fi necesare și alte tipuri mai speciale, rezultând din suprapunerea de unde sinusoidale cu frecvențe diferite, sau având anumite forme de variație, precum segmente liniare cu pante diferite, zgomete gaussiene etc.

În lumea reală, pentru obținerea semnalelor menționate, există generatoare de semnale care furnizează la ieșire tensiuni electrice ce reproduc forma de undă a semnalului selectat. Selectarea tipului de semnal și a caracteristicilor aferente se face, de obicei, cu ajutorul unor butoane, comutatoare, cursoare și alte elemente de interfațare cu operatorul. Vizualizarea semnalului obținut se obține pe ecranul unui osciloscop, iar anumiți parametri, precum frecvența, valoarea efectivă, se pot citi direct pe dispozitive de afișare analogice sau numerice. Prin natura lor, aceste generatoare de semnal pot include într-o singură unitate o gamă relativ redusă de tipuri de semnale și cu posibilități limitate de variație a parametrilor caracteristici. Ca urmare, pentru acoperirea diversității aplicațiilor, pot fi necesare mai multe generatoare, adaptarea lor la anumite cerințe specifice datorită flexibilității restrânse implicând dificultăți și costuri ridicate.

Mediul de programare grafică **LabVIEW** oferă posibilități multiple de creare de **VI-uri** care simulează generarea de semnale, în condițiile în care asigură o mare versatilitate în selectarea tipului de semnal și stabilirea caracteristicilor. Vizualizarea semnalelor se face pe monitorul calculatorului cu **VI-uri** de afișare similare osciloscopului catodic și cu facilități sporite de adaptare a performanțelor în raport de semnal, inclusiv de afișare a mai multor semnale simultan. Astfel, în condițiile mult mai avantajoase ale simulării, se pot stabili algoritmi de sinteză de semnale, se pot verifica și ajusta proceduri de testare simulând inclusiv unitatea supusă testării. Mai mult decât atât, în situația în care se dispune de echipamente de interfață și de comunicație adecvate, semnalele generate prin simulare pot fi convertite în semnale de tensiune utilizabile pentru măsurări și testări în lumea reală, rezultatele pot fi

achiziționate și prelucrate și astfel experimentările pot fi conduse integral cu ajutorul calculatorului.

În pachetul de bază, **LabVIEW** dispune de o serie de module care permit simularea, fără efort de programare, a unor generatoare de semnale având formele de undă reprezentate în fig.4.1, cu următoarea legendă:

- 1 - semnal sinusoidal;
- 2 - semnal rectangular;
- 3 - semnal triunghiular;
- 4 - semnal în dinți de fierăstrău;
- 5 - semnal rampă;
- 6 - semnal impuls

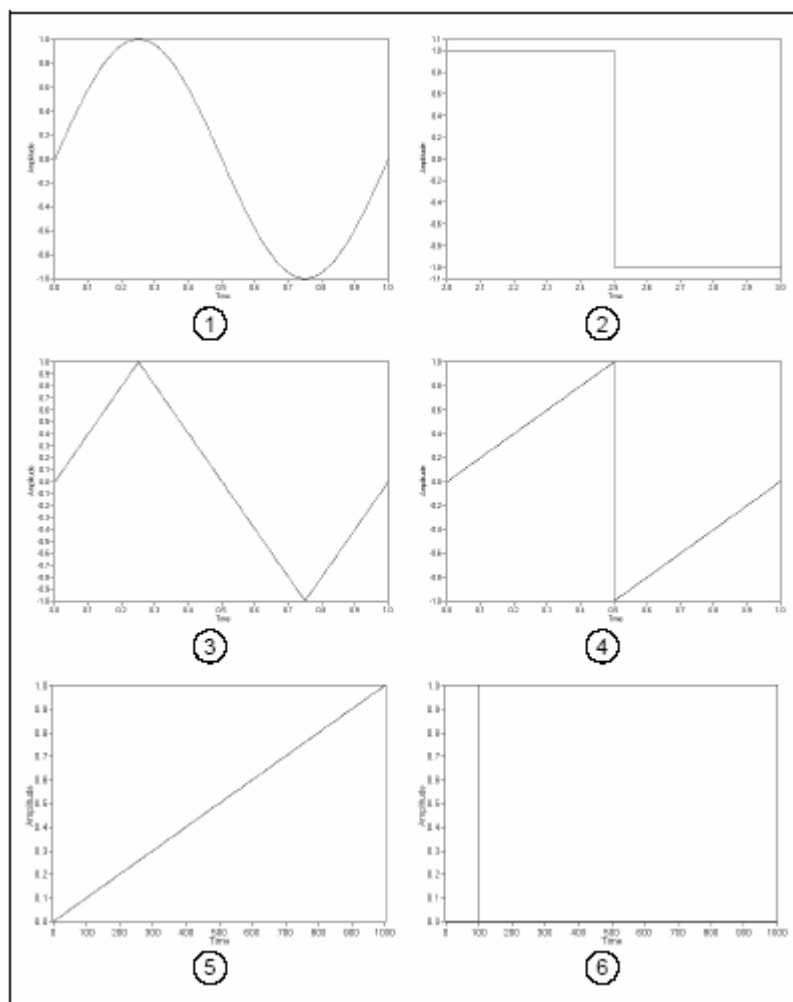


Fig.4.1. Tipuri de forme de undă

Caracterizarea semnalelor din figura precedentă se poate face prin modele în domeniul timpului, deduse direct prin scrierea ecuațiilor corespunzătoare graficelor de mai sus în raport cu variabila t (axa absciselor).

În afara tipurilor de semnale conținute în pachetul de bază, respectiv în paleta **Functions**, există multe alte proceduri de generare în bibliotecile de programe **LabVIEW**, care pot fi accesate din documentații sau cu meniul **Help**.

În cadrul lucrării se vor prezenta și unele modalități de creare de generatoare de funcții, mai precis modalități de generare și trasare a graficelor unor funcții matematice.

Pe parcursul lucrării se vor exemplifica modalități de caracterizare a semnalelor, și - pe această bază - se vor prezenta unele metode de sinteză a semnalelor cu caracteristici prestabilite. Pentru funcții se vor avea în vedere atât unele posibilități de ajustare a parametrilor specifici, cât și altele mai generale, care pot fi comune unor familii de funcții similare.

În toate cazurile exemplificarea se va face on-line prin afișări grafice adecvate.

4.2. Indicatoare pentru vizualizarea semnalelor și a graficelor de funcții

Mediul de programare **LabVIEW** dispune, în meniul **Graph** al paletelor de controale, de o serie de elemente pentru reprezentări grafice. Cele mai utilizate sunt cele aflate pe prima linie a meniului **Graph**: elementele **Waveform Chart** (numit pe scurt **Chart**), **Waveform Graph** (numit pe scurt **Graph**) și **XY Graph**.

În fig.4.2 se pot vedea pictogramele acestor indicatoare, dintre care a fost selectat **Waveform Chart**. Diferențele între cele trei elemente sunt reprezentate de modurile în care acestora li se transmit coordonatele punctelor prin care va fi trasat graficul.

Pentru o mai bună înțelegere a funcționării și utilizării acestor indicatoare, ele vor fi prezentate în asociere cu elemente care furnizează date ce pot reprezenta ordonate ale punctelor unui grafic.

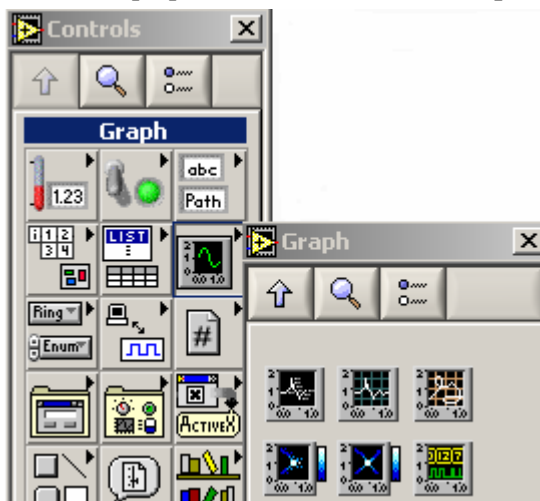


Fig.4.2. Indicatoare grafice pentru semnale

Indicatorul Waveform Chart

Unui element de tip **Chart** i se pot trimite, în mod succesiv, punct cu punct, valorile ordonatelor punctelor pe care să le traseze grafic. Pentru exemplificare, în fig.4.3, este reprezentat un generator de numere aleatoare. Bucla **While** execută câte o iterație la fiecare trei secunde (datorită funcției **Wait (ms)**). La fiecare iterație, funcția

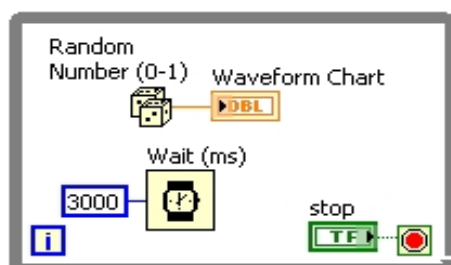


Fig.4.3. Generator de numere aleatoare

Random Number (0-1) trimite câte o valoare aleatoare la elementul **Chart**, care consideră valorile succesive pe care le primește drept ordinate ale punctelor de pe grafic. Abscisele punctelor sunt considerate automat crescătoare din unitate în unitate (0 pentru prima valoare primită, 1 pentru a doua ș.a.m.d.).

Atunci când primește o nouă valoare, un element **Chart** trasează un segment de dreaptă din punctul cel mai recent (ultimul de pe grafic) până în punctul determinat de noua valoare primită.

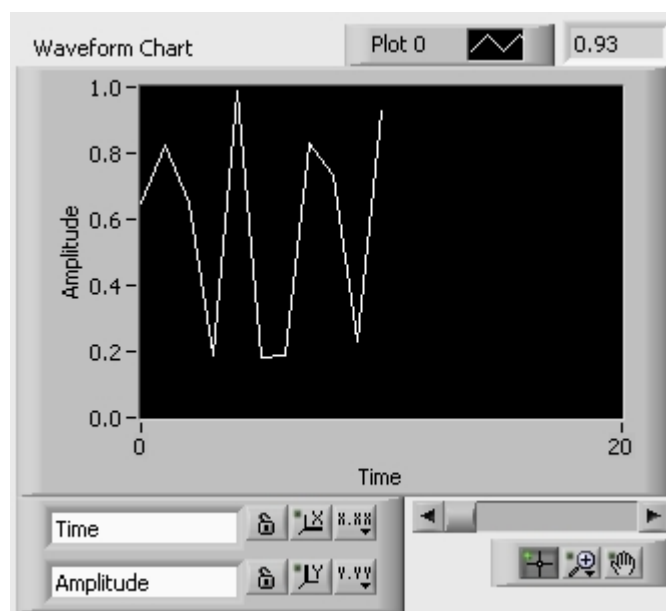
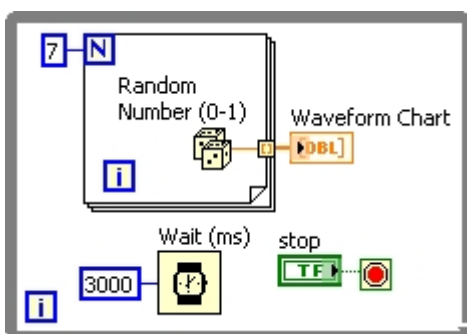
Fig.4.4. Indicatorul **Waveform Chart**

Fig.4.5. Generarea de șiruri de numere aleatoare

În fig.4.4 este redat indicatorul **Waveform Chart** care afișează graficul corespunzător numerelor aleatoare generate.

Un indicator **Chart** acceptă nu numai valori trimise punct cu punct (scalare) ci și șiruri de valori (**Array**). Atunci când primește un șir de valori (considerate, de asemenea, tot ordonate ale punctelor), un element **Chart** adaugă la sfârșitul graficului deja existent nu un singur punct ci un număr de puncte egal cu numărul de valori din șirul primit. În fig.4.5, generatorul de numere

aleatoare, la fiecare iterație a buclei **While** (la fiecare trei secunde), bucla **For** generează un șir (**Array**) de 7 valori aleatoare, șir ce este trimis elementului **Chart**. La fiecare trei secunde, la graficul afișat de **Chart** sunt adăugate astfel încă șapte puncte.

Dacă un element **Chart** primește o matrice de valori numerice (**Array** cu două dimensiuni), atunci el va trasa simultan un număr de grafice egal cu numărul de linii ale matricei.

Indicatorul Waveform Graph

Un element de tip **Graph** nu acceptă valori individuale (scalare) ci numai șiruri (**Array**) de valori. Spre deosebire de elementele **Chart**, atunci când primește un șir de valori, elementul **Graph** șterge graficul pe care îl afișase anterior și afișează doar graficul format din noile puncte primite. În fig.4.6 se poate vedea imaginea indicatorului **Waveform Graph**. Graficul afișat corespunde șirurilor de valori provenite de la un generator, precum cel din fig.4.5. Graficul se va schimba - și în acest caz - la fiecare trei secunde dar, spre deosebire de cel afișat de **Waveform Chart**, șirul este constituit din 7 valori aleatoare. La afișarea unui șir nou valorile afișate anterior se vor pierde.

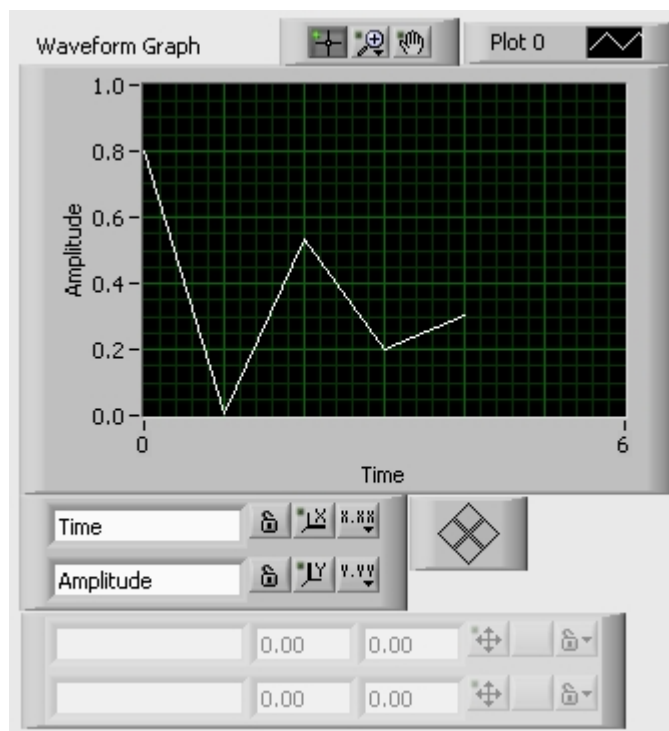
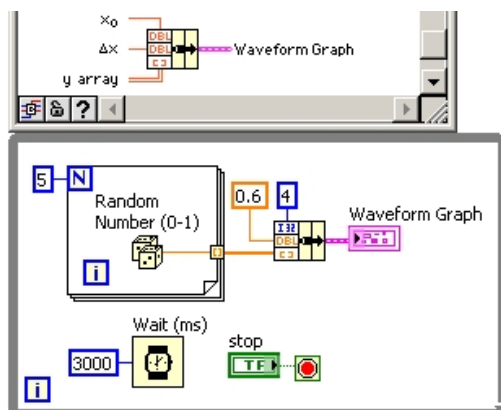


Fig.4.6. Indicatorul **Waveform Graph**

Un element de tip **Graph** consideră, de asemenea, valorile succesive pe care le primește drept ordonate ale punctelor de pe grafic. În modul implicit, abscisele punctelor sunt considerate automat crescătoare din unitate în unitate (0 pentru prima

Fig.4.7. Definirea valorilor x_0 și Δx

(Array cu două dimensiuni), atunci el va trasa simultan un număr de grafice egal cu numărul de linii ale matricei. În situația în care se trasează mai multe grafice și se dorește definirea abscisei x_0 a primului punct și a distanței Δx pe orizontală dintre două puncte succesive, mărimile respective vor fi aceleași pentru toate graficele. În funcția **Bundle**, la cea de a treia intrare se conectează matricea de valori.

valoare primită, 1 pentru a doua ș.a.m.d.). Elementele de tip **Graph** permit însă, suplimentar, definirea abscisei primului punct și a distanței pe orizontală dintre două puncte succesive. În această situație, valorile x_0 și Δx se introduc (printr-o funcție **Bundle**) într-un **Cluster** împreună cu șirul de valori ce reprezintă ordonatele punctelor, iar la terminalul elementului **Graph** se conectează ieșirea funcției **Bundle**, așa cum se poate observa în fig.4.7.

Dacă un element **Graph** primește o matrice de valori numerice

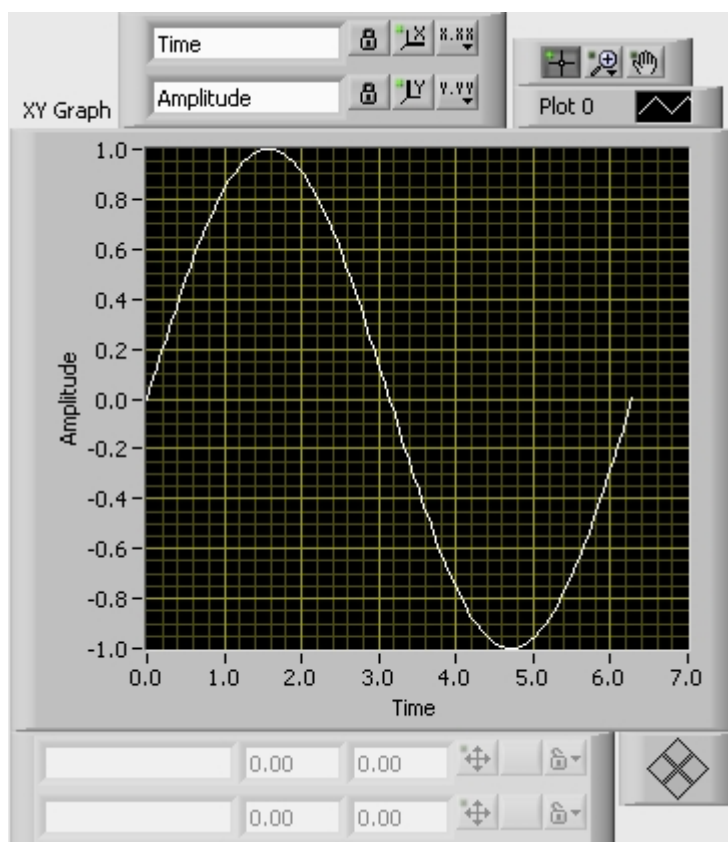


Fig.4.8. Indicatorul XY Graph

Indicatorul XY Graph

Indicatorul de tip **XY Graph** acceptă la intrare un **Cluster** format din două șiruri (**Array**) de valori. Primul șir reprezintă abscisele, iar cel de-al doilea ordonatele punctelor de pe grafic. Evident, cele două șiruri trebuie să conțină același număr de valori. Atunci când se primește o astfel de structură de date, graficul anterior este șters. Imaginea unui indicator **XY Graph** este redată în fig.4.8.

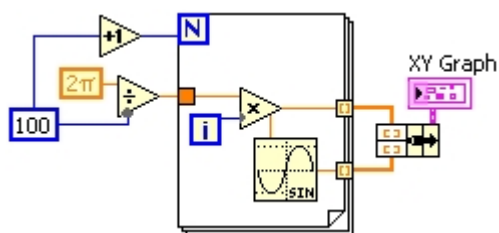


Fig.4.9. Diagrama bloc pentru generarea graficului funcției sinus

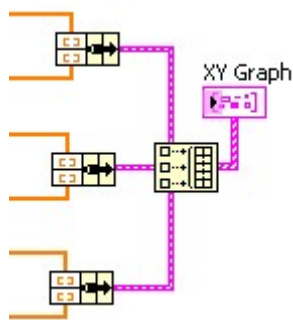


Fig.4.10. Diagrama pentru afișarea mai multor grafice

Graficul afișat este cel al unei sinusoide. În fig.4.9 este arătat un program pentru trasarea graficului funcției sinus între 0 și 2π prin 101 puncte.

Stabilind că punctele vor fi la distanțe egale pe orizontală, se determină întâi distanța Δx pe abscisă dintre două puncte succesive, împărțind la 100 intervalul de trasare, de lungime 2π . Într-o buclă **For** cu numărul de iterații egal cu numărul de puncte, se determină la fiecare iterație abscisa unui punct cu relația $x_i = i \cdot \Delta x$ și se calculează ordonata $y = \sin(x_i)$. La ieșirea din bucla **For**, șirurile de valori x_i și y_i sunt grupate într-un cluster (cu funcția **Bundle**), iar acesta este trimis la terminalul elementului **XY Graph**.

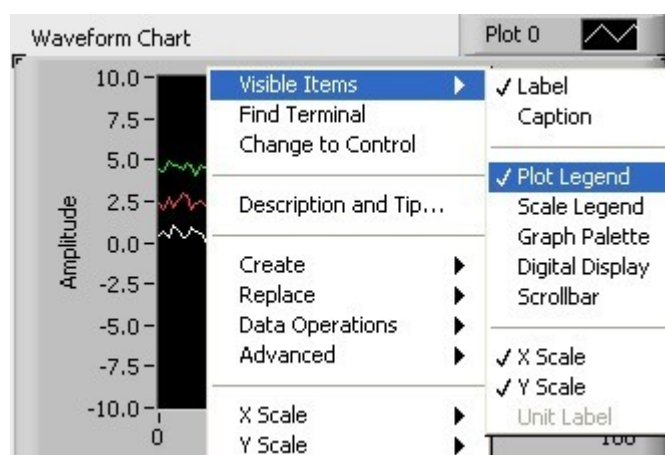
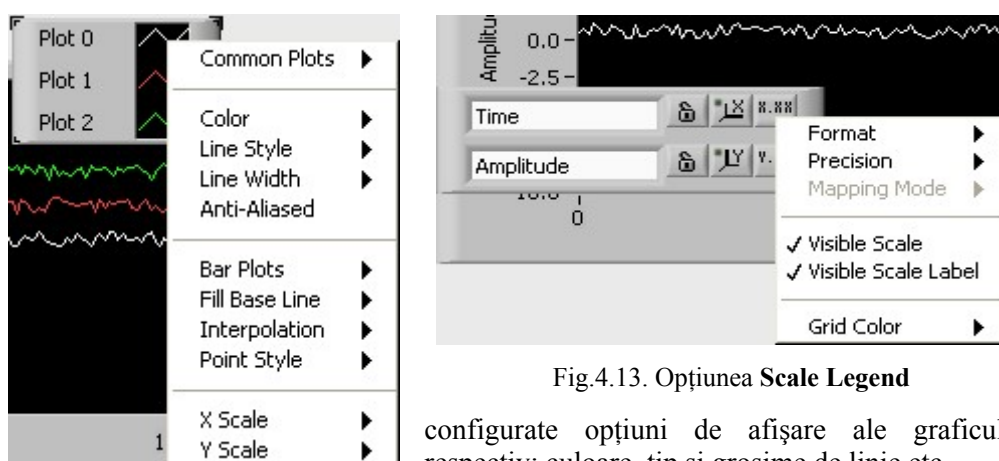
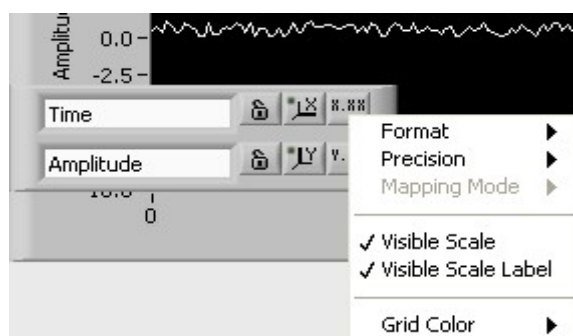
Dacă se dorește trasarea simultană a mai multor grafice se construiește pentru fiecare grafic câte un **Cluster** format din două **Array**-uri (unul pentru abscisele și altul pentru ordonatele punctelor). Ieșirile funcțiilor **Bundle** se conectează la o funcție **Build Array**, iar ieșirea funcției **Build Array** se conectează la terminalul elementului **XY Graph**. Diagrama corespunzătoare este prezentată în fig.4.10.

Opțiuni proprii indicatoarelor grafice

În meniurile proprii ale indicatoarelor grafice se află sub-meniul **Visible Items**, care conține o serie de opțiuni conform listei din fig.4.11, prin intermediul cărora se pot obține facilități de afișare.

Selectând opțiunea **Plot Legend**, ca pe figura precedentă, se afișează o legendă, ca în fig.4.12, ce poate fi poziționată independent, sau dimensionată pentru a avea un număr de poziții egal cu numărul de grafice reprezentate. Numele graficelor (inițial **Plot 0**, **Plot 1** s.a.m.d.) pot fi modificate cu ajutorul uneltei de editare a textelor.

Apăsând butonul drept al mouse-ului atunci când cursorul este poziționat deasupra unui grafic din legendă, se deschide un meniu propriu din care pot fi

Fig. 4.11. Opțiunile din submeniul **Visible Items**Fig.4.12. Opțiunea **Plot Legend**Fig.4.13. Opțiunea **Scale Legend**

configurate opțiuni de afișare ale graficului respectiv: culoare, tip și grosime de linie etc.

Opțiunea **Scale Legend**, din același submeniu și redată în fig.4.13, afișează o componentă ce permite configurarea modului în care sunt afișate scalele indicatorului de reprezentare grafică: eticheta scalei, formatul și precizia de reprezentare, vizibilitatea scalei sau a etichetei acesteia, culoarea carioajului.

Pentru fiecare scală sunt disponibile: un buton (notat cu **X** sau **Y**), la a cărui apăsare se realizează o autoscalare pe direcția respectivă, un buton (marcat cu un lacăt) care, atunci când este apăsă, menține continuu autoscalarea pe direcția corespunzătoare. De regulă, pentru axa **X** se afișează **Time**, având semnificația de axă a timpului, ceea ce este util pentru semnalele care evoluează în raport cu timpul, dar se pot asocia și alte semnificații. O altă opțiune **Graph Palette**, vizibilă în fig.4.14, afișează o componentă cu unelte ce permit: deplasarea graficelor în interiorul elementului (butonul cu mânuță), deschiderea unui submeniu cu unelte pentru mărire sau micșorare statică sau dinamică pe diverse direcții.

Fig.4.14. Opțiunea **Graph Palette**

Există și unele opțiuni specifice fiecăruia dintre cele trei tipuri de indicatoare grafice descrise, opțiuni care se găsesc în meniurile proprii.

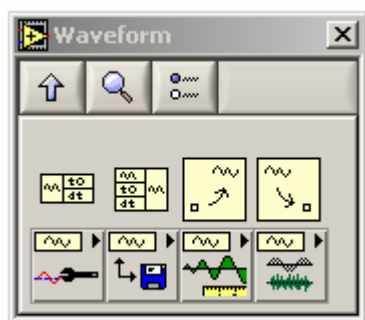
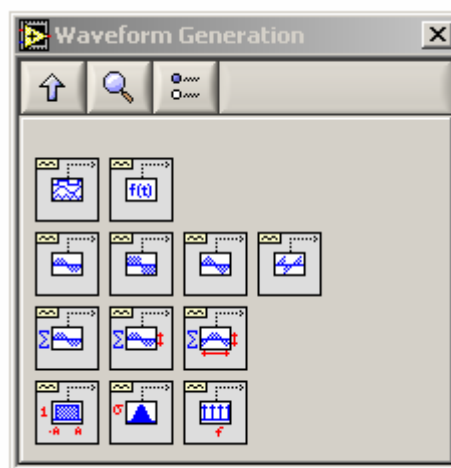
4.3. Proceduri și instrumente pentru simularea generării semnalelor

Crearea de VI-uri care să simuleze generarea de semnale se poate face, în **LabVIEW**, în următoarele moduri:

- folosind module dedicate aflate în pachetul de bază (**Templates**);
- combinând diverse module de funcții din bibliotecile **LabVIEW**.

4.3.1. Simularea generării folosind module dedicate

Deschizând o nouă aplicație, în paleta **Functions** se execută click pe subpaleta **Waveform** și aceasta este afișată ca în fig.4.15.

Fig.4.15. Subpaleta **Waveform**Fig.4.16. Fereastra **Waveform Generation**

Selectând apoi icon-ul **Waveform Generation** (colțul din dreapta jos) se deschide fereastra din fig.4.16.

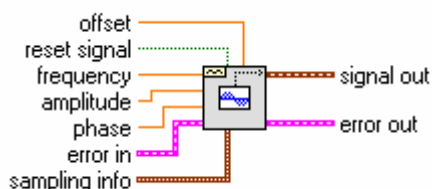
După cum se poate observa pe fig.4.16, în fereastra respectivă apar mai multe icon-uri, care reprezintă module generatoare de forme de undă. Astfel, cu referire la cele ce vor fi utilizate în lucrare, pe rândul al doilea de sus se află 4 module care,








corespunzător desenelor afișate, pot genera următoarele forme de undă: sinusoidală, rectangulară, triunghiulară și în dinți de fierăstrău. Executând click pe unul dintre module, acesta se poate trage în diagrama bloc și constituie nucleul în jurul căruia poate fi creat generatorul virtual pentru forma de undă dorită.

Pentru a înțelege utilizarea acestor module, se vor prezenta, în continuare, terminalele cu care sunt dotate și tipurile de elemente cu care se interconectează.

4.3.2. Generator bazat pe modulul Sine Waveform

Este modulul pentru generarea unei sinusoidale și are terminalele din fig.4.17.



-  - **offset**, este componenta continuă, valoarea predefinită **0**.
-  - **reset signal**, resetează faza conform controlului **phase**, valoarea predefinită **False**.
-  - **frequency**, este frecvența semnalului în Hz, valoarea predefinită **10**.
-  - **amplitude**, este amplitudinea sinusoidei, valoarea predefinită **1**.
-  - **phase**, este faza inițială în grade, valoarea predefinită **0**.
-  - **error in**, exprimă eroarea inițială, valoarea predefinită **no error**; acest control se găsește în subpaleta **Array&Cluster** și uneori poate fi omis.
-  - **sampling info**, conține informații despre frecvența de eșantionare **F_s** și numărul de eșantioane **#s**, incluzând controalele de mai jos ale căror valori predefinite sunt 1000 (pentru ambele)





 **F_s**
 **#s**
-  - **signal out**, este forma de undă generată;
-  - **error out**, conține informații cu privire la erorile care pot apare pe parcursul rulării aplicației; ca și **error in** se ia tot din subpaleta **Array#Cluster**.

Fig.4.17. Schema și terminalele modulului **Sine Wave**

Pe fig.4.17 sunt arătate și terminalele din diagrama bloc ale controalelor și indicatoarelor din panoul frontal, care se conectează cu modulul **Sine Waveform**, rezultând astfel informațiile necesare asupra configurării panoului frontal.

Din fereastra **Waveform Generation** se pot accesa module asemănătoare și pentru celelalte forme de undă. Informații de natura celor din fig.4.17, pentru oricare

dintre modulele afișate în această fereastră, se obțin folosind meniul **Help** din care se selectează opțiunea **VI, Function & How To Help** și apoi în spațiul de căutare se scrie denumirea modulului dorit, de exemplu **Sine Wave/Square Wave/Triangle Wave ...etc.** Apoi se execută click pe **List Topics** și apar o serie de adrese, între care și cea care conține modulul căutat. Efectuând click la poziția respectivă se afișează o pagină precum cea din fig.4.17, care conține și alte date suplimentare.

Un generator virtual de semnal sinusoidal se obține prin asocierea modulului **Sine Wave** cu controalele și indicatoarele precizate anterior. Panoul frontal al unui astfel de VI este redat în fig.4.18.

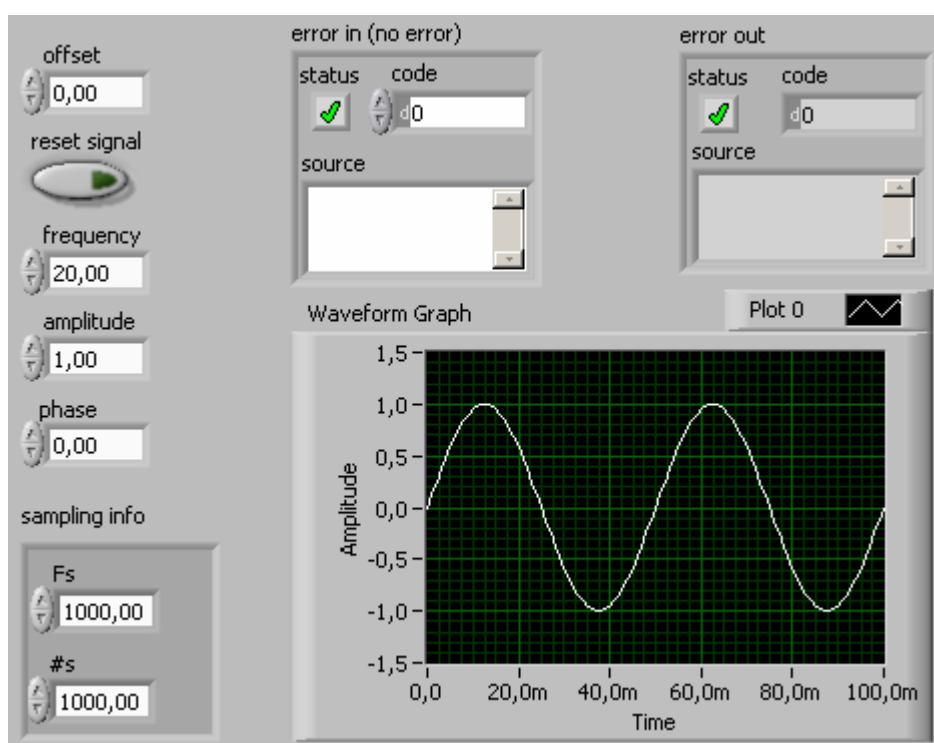


Fig.4.18 Panoul frontal al generatorului de semnal sinusoidal

Pe figură se pot observa controalele numerice pentru adăugarea unei componente continue (offset), pentru frecvență, pentru amplitudine și pentru faza inițială a semnalului. De asemenea, se mai găsește un control de tip **array** pentru frecvența de eșantionare **F_s** și pentru numărul de eșantioane **#s** din care este constituit semnalul. Se mai poate vedea și un control boolean necesar pentru resetare. Referitor la erori (**error in** și **error out**), în panou sunt plasate un control și, respectiv, un indicator extrase din paleta **Controls** subpaleta **Array & Cluster**. În sfârșit, pentru vizualizarea semnalului generat, este prevăzut un indicator denumit **Waveform Graph** care se află în paleta **Controls** subpaleta **Graph**. Despre acest tip de indicator se găsesc detalii în secțiunea 4.2.

În fig.4.19 este reprezentată diagrama bloc a generatorului virtual de semnal sinusoidal.

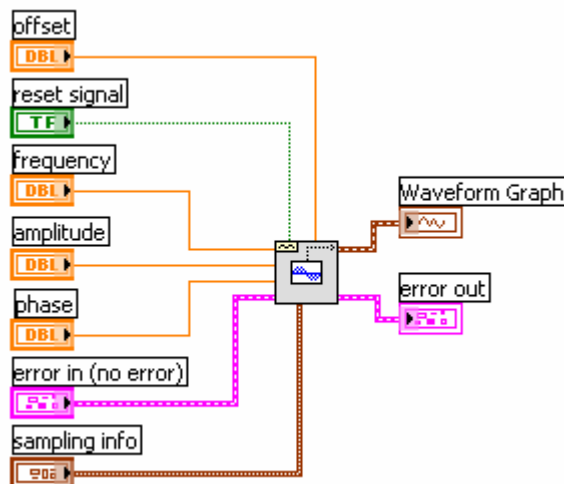


Fig.4.19. Diagrama bloc a generatorului virtual de semnal sinusoidal

Pentru ușurința în trasarea conexiunilor, se plasează pointer-ul mouse-ului peste icon-ul **Sine Wave** și se execută click dreapta, astfel încât să se afișeze meniul propriu din care se alege opțiunea **Visible Items**. Din fereastra care se deschide, se selectează **Terminals**, ceea ce are ca efect schimbarea imaginii modulului, astfel încât să fie vizibile locațiile terminalelor la care trebuie conectate controalele și indicatoarele precizate în fig.4.17. La plasarea unei etichete de conectare peste un terminal, pentru scurt timp apare denumirea acestuia și poate fi identificat.

După terminarea operațiilor sus-menționate și fixarea valorilor predefinite ale controalelor se poate trece la rularea aplicației cu **Run Continuously**. Pe indicatorul grafic se afișează una sau mai multe perioade ale unei sinusoidale cu amplitudinea **1** și faza inițială **0**. Numărul de perioade depinde de modul în care este gradată axa **X**, axa timpului. Dacă la capătul acestei axe se află valoarea **1,0**, ceea ce semnifică o secundă, se vor afișa 10 perioade, care vor ocupa întreaga axă **X**. Explicația constă în faptul că valorile predefinite sunt: $F_s = 1000\text{Hz}$ și $\#s = 1000$, iar perioada semnalului $T = 1/10 = 0,1\text{ s}$. Gradarea axei **X** se poate modifica pentru a vizualiza numărul de perioade dorit. Pentru aceasta se va verifica mai întâi deselectarea opțiunii **Auto Scale X** din meniul propriu al indicatorului **Waveform Graph**. Apoi, cu unealta **Edit Text** din paleta **Tools**, se șterge valoarea existentă la capătul axei **X** și se tipărește valoarea dorită. Executând apoi click în afară se modifică și restul gradațiilor axei. Pentru ușurință, trebuie avute în vedere următoarele relații:

- durata afișării semnalului: $t_a = \#s/F_s$
- numărul de perioade afișate: $n_a = t_a/T$, unde T este perioada semnalului.

Modificarea duratei de afișare și a numărului de perioade afișate se poate face și prin schimbarea frecvenței de eșantionare F_s și a numărului de eșantioane $\#s$. În acest caz trebuie să se țină seama de teorema eșantionării (Shannon), astfel ca să

rezulte un anumit număr de eșantioane pentru o perioadă a semnalului. Numărul minim de eșantioane, pentru ca semnalul să fie relativ neted este, de regulă, **100**. Cu cât numărul de eșantioane pe perioada $np = T \cdot F_s$ este mai mare, cu atât semnalul generat este mai neted. Pornirea și oprirea rulării programului se face numai de pe bara de comenzi, cu **Run Continuously**, respectiv **Stop**.

4.3.3. Generator de semnale multiple folosind module de tipuri diferite

Utilizând module de tipuri diferite asociate cu controalele și indicatoarele menționate în exemplul anterior, se va arăta - în continuare - crearea unui generator virtual, care să permită generarea a 4 forme de undă: sinusoidală, triunghiulară, rectangulară și în dinți de fierăstrău. Se va putea constata faptul că selectarea semnalului dorit se poate face ușor, ca și trecerea de la un tip de semnal la altul.

În fig.4.20 este redat panoul frontal al generatorului de semnale multiple. Se poate observa că se aseamănă cu cel din fig.4.18 creat pentru semnalul unic sinusoidal, dar apar și unele deosebiri.

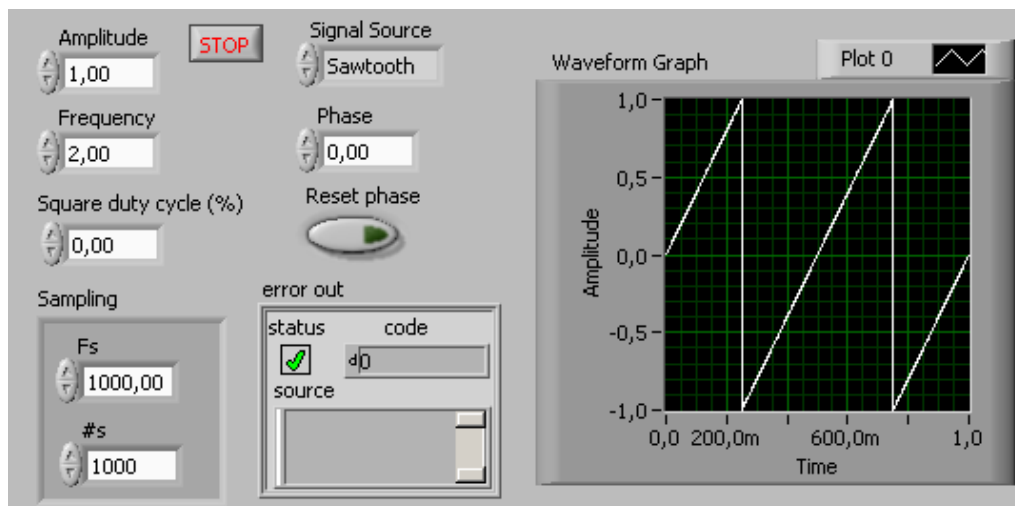


Fig.4.20. Panoul frontal al generatorului de semnale multiple

Astfel, pentru simplificare, s-a renunțat la controlul pentru **offset** și la cel pentru **error in**. În schimb s-a introdus controlul numeric **square duty cycle** necesar pentru semnalul **Square Wave** și controlul boolean **Stop**, prin intermediul căruia se condiționează structura repetitivă **While Loop**, care se va vedea că apare în diagrama bloc. De asemenea, s-a mai adăugat controlul de tip enumerativ **Signal Source** preluat din paleta **Controls**, subpaleta **Ring&Enum**, selectând **Enum**. Controlul enumerativ a permis introducerea listei cu cele 4 semnale, care vor fi generate și care pot fi selectate acționând asupra săgeților cu unealta **Operate Value** din paleta **Tools**. Semnalul selectat apare afișat în eticheta atașată acestui control; în fig.4.20 se vede **Sawtooth**.

Diagrama bloc a generatorului de semnale multiple se poate vedea în fig.4.21.

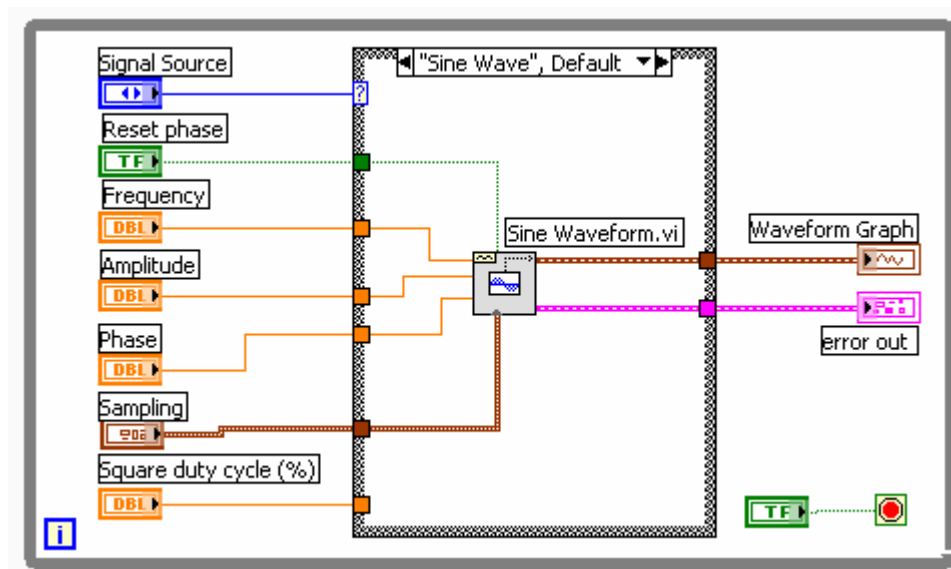


Fig.4.21. Diagrama bloc a generatorului de semnale multiple

La fel ca și în cazul panoului frontal, o serie de elemente sunt aceleași ca în cazul diagramei bloc pentru generatorul sinusoidal din fig.4.19, dar apar și deosebiri semnificative. În figura precedentă diagrama bloc este cea care corespunde generării semnalului sinusoidal, acesta fiind cazul predefinit (Default), care apare la deschiderea aplicației. Se poate observa că modulul **Sine Wave** se află în interiorul unei structuri **Case**, care se dimensionează astfel încât restul de elemente, controale și indicatoare să rămână în exteriorul acesteia. Structura cauzală **Case** execută subdiagrama din interiorul ei în funcție de specificația care se află înscrisă în eticheta de pe latura superioară. Această inscripție este determinată de controlul enumerativ **Signal Source**, al cărui terminal este conectat la terminalul selector (de cazuri) aflat pe conturul structurii, în cazul de față va fi **Sine Wave**, așa cum arată controlul corespunzător din panoul frontal. Săgețile din eticheta care afișează numele forme de undă permit o derulare din care să se vadă toate tipurile care pot fi generate. Pe figură se pot observa conexiunile controalelor și indicatoarelor care străbat frontiera structurii **Case** prin intermediul terminalelor (pătratelor) de pe contur și, în interiorul acesteia, sunt conectate la modulul **Sine Wave** la fel ca în cazul generatorului sinusoidal (fig.4.19). Terminalul **Square Duty Cycle** nu este conectat, el fiind specific undei rectangulare.

Întreg ansamblul descris anterior este înglobat într-o structură repetitivă cu condiție de terminare **While Loop**, care are proprietatea de a executa subdiagrama pe care o conține până când la terminalul de continuare (colțul din dreapta jos) apare o anumită valoare booleană, furnizată de butonul **Stop** din panoul frontal. De exemplu, dacă terminalul se află în starea implicită **Continue if True**, structura va continua să execute iterații succesive atât timp cât la terminalul său de continuare ajunge o valoare logică **True** și încetează (nu mai trece la următoarea iterație) dacă ajunge o valoare logică **False**. Starea implicită a terminalului de continuare se poate modifica

executând click dreapta deasupra sa și deschizând meniul său propriu. Se poate opta, de exemplu, pentru **Stop if True**.

Structuri **Case** similare cu cea din fig.4.21 se crează și pentru celelalte tipuri de semnale ce urmează a fi generate. În fig.4.22 sunt redată aceste structuri.

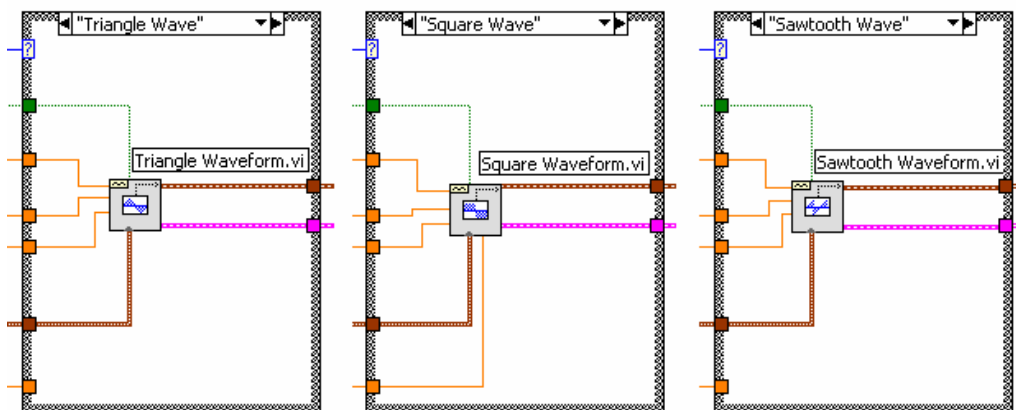


Fig.4.22. Structurile **Case** pentru modulele **Triangle Waveform**, **Square Waveform** și **Sawtooth Waveform**

Pentru ca structurile din fig.4.22 să se poată insera succesiv în diagramă în locul structurii **Sine Waveform** din fig.4.21, trebuie să se efectueze următoarele operații:

- se alege cu controlul **Signal Source** din panoul frontal tipul semnalului, de exemplu **Triangle Waveform** și se observă ca în diagramă eticheta structurii **Case** să aibă aceeași inscripție (dacă nu este așa se folosesc săgețile);
- se execută click dreapta pe structura **Case** din fig.4.21 și, din meniul propriu, se selectează **Add Case After**;
- în același mod se procedează și pentru structurile corespunzătoare celorlalte module, astfel încât, în final, manevrând săgeata verticală din eticheta structurii **Case**, să se alinieze lista cu cele 4 semnale. În acest mod, inserarea structurii **Case** corespunzătoare se face automat, în momentul selectării semnalului dorit cu controlul **Signal Source**, și acesta poate fi vizualizat pe indicatorul grafic **Waveform Graph**. Parametrii caracteristici pentru oricare dintre cele 4 semnale pot fi ajustați independent din momentul în care semnalul respectiv a fost selectat. Asemănarea cu un generator multisemnal real, la care alegerea tipului de semnal și a parametrilor se face cu butoanele și comutatoarele de pe panou, este evidentă.

4.3.4. Generarea și afișarea grafică simultană a mai multor semnale

În cadrul anumitor operații de măsurare sau de testare, poate să apară necesitatea generării și vizualizării simultane a mai multor semnale de tipuri și cu caracteristici diferite. Posibilitățile pe care le oferă programele **LabVIEW** în acest sens vor fi ilustrate în cele ce urmează.

Aplicația va consta în generarea și afișarea simultană a două semnale, unul sinusoidal și altul triunghiular, utilizând modulele corespunzătoare **Sine Waveform** și **Triangle Waveform**. Panoul frontal este redat în fig.4.23.

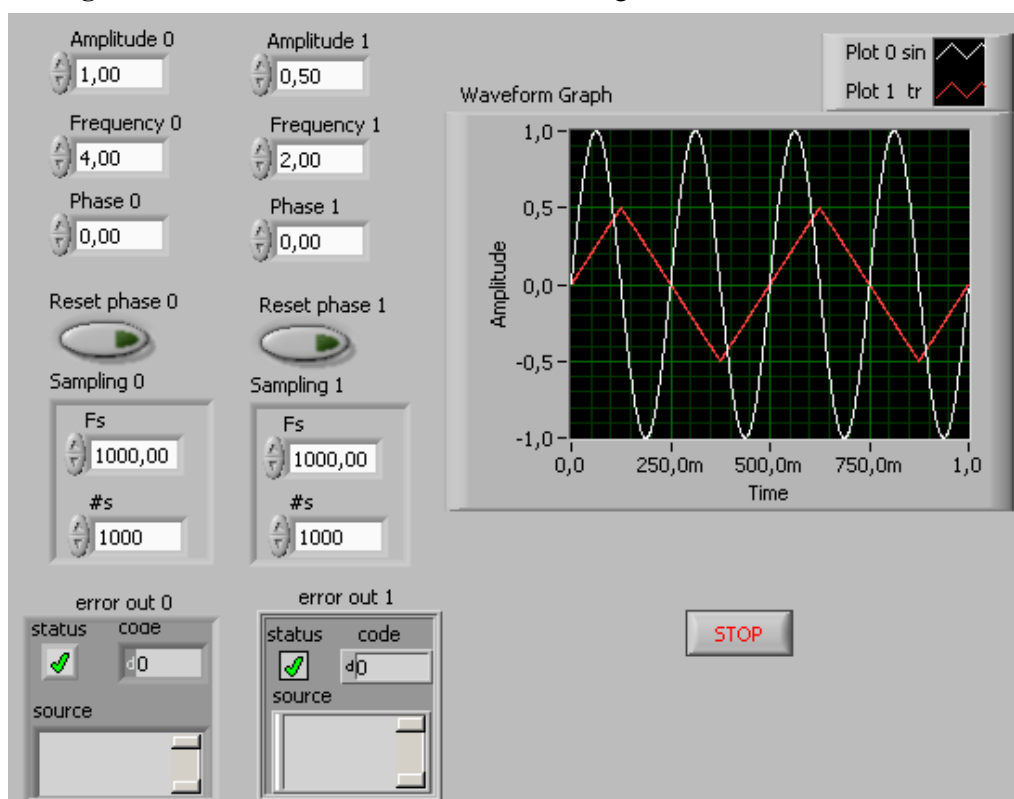


Fig.4.23. Panoul frontal pentru generarea și afișarea grafică a două semnale simultan

Panoul frontal cuprinde, în esență, aceleași controale precum cele din fig.4.18, corespunzătoare unui singur semnal - cel sinusoidal - deosebirea constând în faptul că sunt dublate, astfel încât să se asigure independența în ajustarea parametrilor celor două semnale. Cele notate cu indexul **0** aparțin semnalului sinusoidal, iar cele cu indexul **1** semnalului triunghiular. Ca indicator grafic s-a prevăzut, de asemenea, **Waveform Graph**, care va fi folosit pentru ambele semnale simultan. În acest scop se poziționează cursorul mouse-ului pe colțul dreapta sus, unde este scris **Plot 0**, și cu butonul stâng apăsat, atunci când apar echerile pe colțurile dreptunghiului de încadrare, se mărește suprafața acestuia până se dublează. La eliberarea butonului apare scris **Plot 1**. Cu unealta **Edit Text** din paleta **Tools** se completează denumirile abreviate ale graficelor care vor fi afișate, **sin** alături de **Plot 0** și **tr** lângă **Plot 1**.

Din meniul propriu al indicatorului **Waveform Graph** se deschide **Visible Items** și se pot selecta culorile graficelor, modurile de configurare a scalelor etc, așa cum s-a arătat în secțiunea 4.2 paragraful **Opțiuni proprii indicatoarelor grafice**.

Se mai observă, pe panou, un singur buton **Stop**, întreruperea rulării programului fiind comună pentru ambele semnale.

Diagrama bloc, corespunzătoare panoului frontal, este reprezentată în fig.4.24. Se remarcă existența celor două module, având fiecare terminalele conectate cu cele ale controalelor dispuse în panoul frontal, de o manieră asemănătoare cu cea din fig.4.20. Toate elementele menționate sunt incluse într-o structură **While Loop**, care asigură rularea condiționat de semnalul dat de controlul boolean **Stop**.

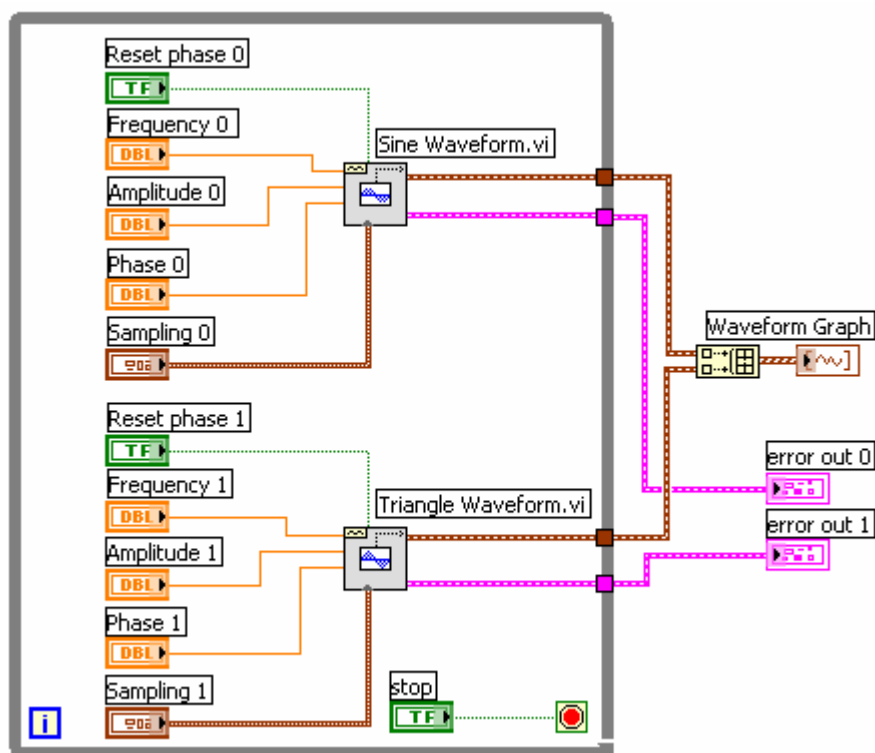


Fig.4.24. Diagrama bloc pentru generarea și afișarea simultană a două semnale

În ceea ce privește ieșirile celor două module, acestea sunt scoase prin tunele create pe frontiera structurii **While Loop** și, prin intermediul unei funcții **Build Array** cu două elemente, sunt conectate la **Waveform Graph** plasat în afara structurii.

Valorile parametrilor semnalelor (amplitudinea, frecvența, faza inițială precum și frecvența de eșantionare și numărul de eșantioane) se stabilesc de la controalele de pe panoul frontal, după care se poate începe rularea cu **Run**. Valorile parametrilor celor două semnale pot fi distincte, dar pentru a obține imagini complete ca amplitudine, stabile și cu un număr convenabil de perioade, trebuie acordată atenție la ajustarea scalelor. Se va ține seama de satisfacerea, pentru ambele semnale, a relațiilor dintre **na**, **ta**, **T**, **Fs** și **#s** expuse în secțiunea 4.3. Oprirea rulării se face utilizând butonul **Stop**.

4.4. Instrumente pentru trasarea graficelor funcțiilor matematice

Programe **LabVIEW** similare celor arătate anterior pentru semnale permit crearea de instrumente virtuale pentru trasarea graficelor funcțiilor matematice. În cadrul lucrării se vor prezenta două proceduri destinate acestui scop:

- trasarea cu ajutorul unor module dedicate existente în pachetul de bază;
- trasarea folosind structuri de tipul noduri cu formule.

4.4.1. Trasarea graficelor de funcții cu module dedicate

Deschizând paleta **Functions** și apoi subpaleta **Numeric**, se observă că, pe lângă icon-urile reprezentând operații aritmetice și unele funcții elementare, există și unele care afișează reprezentări grafice, de exemplu icon-ul de pe ultima coloană din dreapta care conține graficul funcției sinus. Plasând pointer-ul mouse-ului deasupra acestui icon, la partea superioară se afișează denumirea **Trigonometric** și executând click se deschide fereastra din fig.4.25.

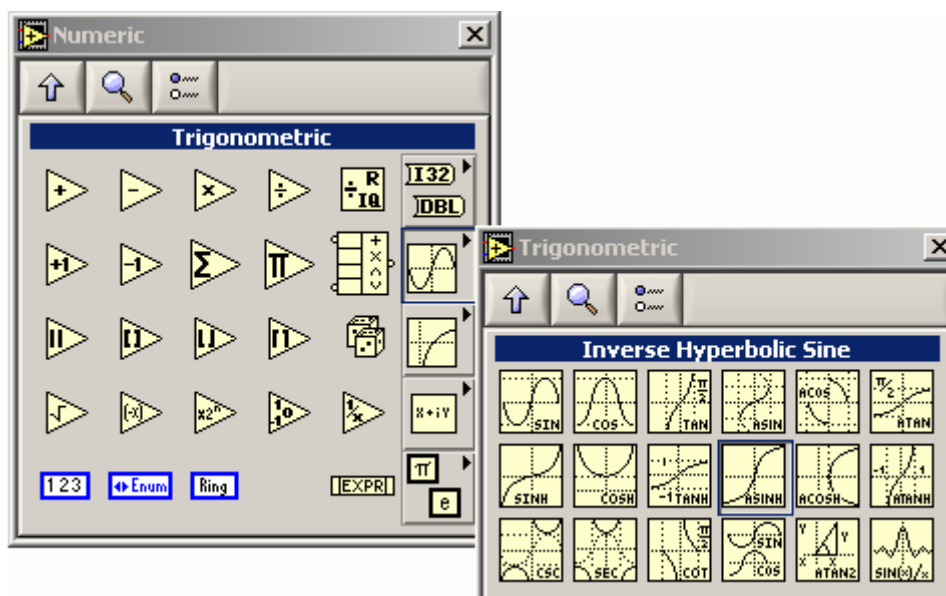
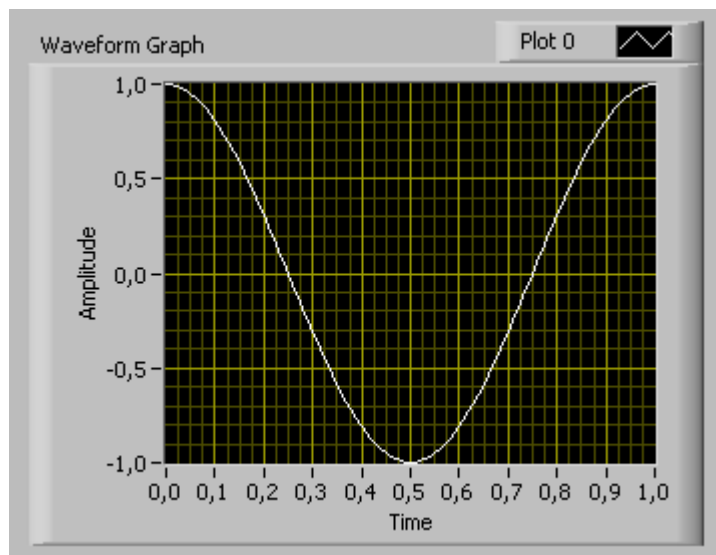


Fig.4.25. Fereastra **Trigonometric**

În fereastra din fig.4.25 se pot observa module care afișează graficele unor funcții trigonometrice directe și inverse și care pot fi folosite pentru crearea de instrumente de trasare a graficelor respective, prin proceduri similare celor de la generarea formelor de undă.

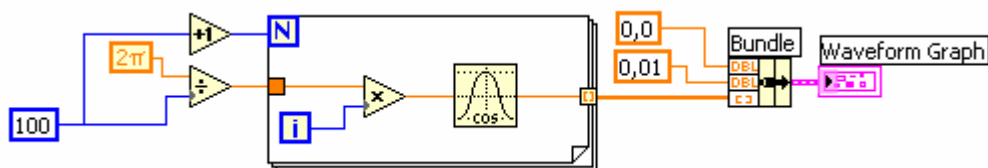
Pentru a ilustra utilizarea acestor module, se deschide o nouă aplicație **LabVIEW** în al cărei panou frontal se plasează un indicator grafic de tipul **Waveform Graph**, ca în fig.4.26.

Scalele axelor sunt gradate convențional, axa **X** pentru abscise pe intervalul $0 \dots 1$ și axa **Y** pentru ordinate pe intervalul $-1 \dots +1$. Deselectând opțiunea **Auto Scale** din meniul propriu al indicatorului, se pot introduce gradațiile dorite. De asemenea, pe

Fig.4.26. Panoul frontal pentru graficul **cosinus**

imaginea din fig.4.26 scalele nu sunt etichetate, lăsându-se la latitudinea utilizatorului de a le eticheta după dorință, folosind opțiuni din același meniu propriu.

În fig.4.27 este reprezentată diagrama bloc a **VI**-ului pentru trasarea graficului funcției **cosinus**. Se observă plasarea în diagramă a unei structuri repetitive **For Loop** în interiorul căreia s-a introdus modulul care afișează graficul unei cosinusoide, preluat prin tragere din fereastra **Trigonometric** arătată în fig.4.25. S-a prevăzut ca bucla **For** să execute **N=100** de iterații, cu increment unitate. În acest scop intervalul de definiție $0 \dots 2\pi$ s-a împărțit la 100, subintervalul elementar fiind transmis printr-un tunel care străbate granița buclei **For** la un element de înmulțire cu numărul **i** al iterației curente. Aceasta permite distribuirea echidistantă și uniformă a eșantioanelor pe intervalul prestabilit. Valorile de la ieșirea modulului **cosinus** sunt transmise în exteriorul buclei **For** tot printr-un tunel practicat pe frontiera acesteia. Prin execuția repetată de către bucla **For** rezultă un **Array** format din valori ale funcției cosinus care se cablează la intrarea **C3** a unui element **Bundle**. La celelalte intrări s-au cablat: x_0 - abscisa începutului graficului și **dx** - distanța între punctele din care este constituit graficul. Pentru a asigura consistența cu împărțirea prealabilă a intervalului de definiție și modul de distribuire a eșantioanelor, s-a ales $x_0 = 0,0$ și **dx=0,01**. Ieșirea **Bundle**-ului este conectată la intrarea indicatorului **Waveform Graph**, care primește astfel toate valorile necesare trasării graficului, intervalul de definiție, punctul de

Fig.4.27. Diagrama bloc pentru trasarea graficului **cosinus**

început, abscisele și ordonatele punctelor.

Aplicația se rulează cu **Run Continuously** și pe indicator apare graficul funcției cosinus trasat pe un interval egal cu o perioadă, desfășurat pe toată întinderea axei **X**. Pornirea și oprirea rulării se face numai de la butoanele de pe bara de comenzi. Parametrii graficului pot fi modificați după necesități, prin schimbarea valorilor din diagrama bloc și a gradării axei **X**.

4.4.2. Trasarea graficelor de funcții utilizând noduri cu formule

Un nod cu formule în **LabVIEW** reprezintă o structură care se găsește în paleta **Functions**. În subpaleta **Structures** se află structura denumită **Formula Node**, cu ajutorul căreia se pot introduce formule matematice direct în diagrama bloc și se pot executa operațiile pe care acestea le conțin. În mod asemănător cu oricare dintre structurile cu care s-a lucrat anterior, se trage și această structură în diagrama bloc și i se ajustează dimensiunile în raport de formula care urmează a fi utilizată. Apoi, cu unealta **Edit Text** din paleta **Tools**, se scrie în interiorul structurii respective formula sau ecuația care se dorește a fi calculată. Sintaxa utilizată pentru scrierea formulelor matematice este similară cu sintaxa utilizată de limbajele de programare textuale. Pentru eventuale lămuriri se poate folosi meniul **Help** și în cadrul acestuia se caută **Creating Formulas >> Formula Nodes Syntax**.

Pentru ilustrarea utilizării nodurilor cu formule, în prezenta lucrare s-a ales relația $y=A*\sin(i*x)$, în ideea trasării graficului unei sinusoide. Conform celor menționate mai sus, această relație se scrie în interiorul structurii. Trebuie subliniat că în cazul în care expresia care trebuie executată conține mai multe egalități, care pot defini unele variabile intermediare, acestea vor fi scrise pe linii separate, iar sfârșitul fiecărei linii va fi marcat cu semnul **;**. Pentru a putea introduce variabilele independente de intrare în formule, pe frontiera structurii se crează tunele prin executare click dreapta și selectarea opțiunii **Add Input**. În cazul sinusoidei intrările sunt **A** și **x**, simboluri care se vor înscrie în terminalele create. În mod asemănător se

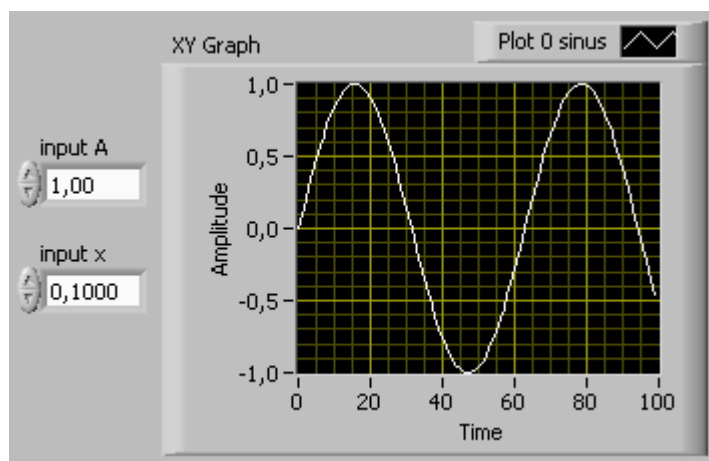


Fig.4.28. Elemente din panoul frontal al aplicației nodului cu formule pentru trasarea graficului sinusoidei

vor crea terminale pentru mărimile de ieșire, de regulă, pe latura opusă intrărilor. Se execută click dreapta și se selectează **Add Output**. Pentru sinusoidă mărimea de ieșire este y , simbol care se va înscrie în terminalul de ieșire. Operația următoare constă în cablarea terminalelor controalelor și indicatoarelor plasate în diagrama bloc cu cele create pe frontiera structurii **Formula Node**. Toate operațiile menționate sunt ilustrate pe fig.4.28 și fig.4.29, care reprezintă elemente din panoul frontal și din diagrama bloc pentru această aplicație.

Cu scopul de a obține facilități de trasare a graficului, structura **Formula Node** este plasată în interiorul unei structuri repetitive **For Loop**. La contorul care indică numărul de iterații s-a înscris numărul **100** urmat de simbolul care specifică

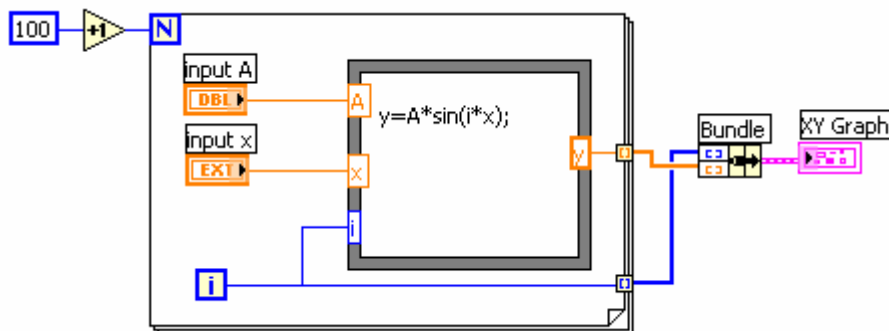


Fig.4.29. Diagrama bloc pentru trasarea graficului sinusoidelor

incrementul, în cazul de față incrementul este **1**. Observând că graficul va fi construit dintr-un număr de puncte egal cu numărul de iterații, numărul i corespunzător iterației curente trebuie introdus în formulă ca variabilă de intrare și, ca urmare, se va crea un nou terminal de intrare pe frontiera structurii **Formula Node**, procedând la fel ca și pentru celelalte. Indicatorul grafic de tip **XY**, plasat în exteriorul buclei **For** este conectat printr-un **Bundle** la cele două șiruri de valori (**array-uri**), unul constituit din ordonatele graficului reprezentate de valorile y ale ieșirii și celălalt din abscisele graficului date de valorile i ale iterațiilor. Ambele șiruri de valori sunt scoase la intrările **Bundle**-ului prin tunele practicate pe frontiera buclei **For**.

Ca și în alte aplicații, trebuie avut grijă ca terminalele care se conectează să aibă același tip de reprezentare și de precizie. O altă regulă generală pentru nodurile cu formule este aceea că, în mod obligatoriu, toate terminalele de intrare trebuie conectate, pe când unele dintre cele de ieșire pot rămâne libere. Mai trebuie precizat că variabilele intermediare pot fi declarate ca ieșiri, nu este necesar ca ele să fie scoase și nominalizate pe frontiera structurii de tip nod cu formule.

După crearea panoului frontal și a diagramei bloc, conform celor arătate anterior, se introduc valorile de intrare - în acest caz $A=1$ și $x=0,1000$ - după care aplicația poate fi rulată cu **Run Continuously**. La terminarea celor 100 de iterații, pe ecranul indicatorului apare graficul din fig.4.28. Se observă că s-a trasat ceva mai mult de 1,5 perioade ale sinusoidelor cu amplitudinea 1, ceea ce corespunde variabilelor de intrare introduse. Pentru amplitudine, aceasta se vede direct din gradarea axei Y . În

ceea ce privește întinderea graficului pe axa **X**, gradată în număr de iterații, acesta se întinde de la valoarea inițială a argumentului $i.x = 0$, pentru $i=0$, până la $100.x$ la finele iterațiilor, adică $100.0,1=10$. Ținând cont că unei perioade îi corespund 2π radiani și calculând $10: 6,28$ rezultă 1,59 perioade. Dacă se modifică valorile capetelor de scală pentru cele două axe, forma graficului se va schimba corespunzător. În cazul nodurilor cu formule, variabilele sunt considerate scalari numerici fără a avea atașate unități de măsură. Unitățile de măsură pot rezulta indirect din modul cum sunt gradate cele două axe și de semnificația variabilelor respective în cadrul relațiilor de calcul.

NOTĂ: Toate **VI**-urile prezentate în lucrare se află salvate în directorul **My Documents** >> **Aplicatia II.4**.

4.5. Chestiuni de studiat

1. Se vor deschide și se vor experimenta toate **VI**-urile din directorul **My Documents** >> **Aplicatia II.4** referitoare la generarea de semnale și trasarea de grafice.

2. Se va realiza un program în cadrul căruia, pe un indicator de tip **Waveform Chart**, se vor trasa două grafice cu valori aleatoare, unul având valori între 0 și 1 și celălalt cu valori între 2 și 3.

3. Se va repeta tematica precedentă în cazul utilizării unui indicator de tipul **Waveform Graph**. Se va face o comparație între modurile de afișare pentru **Waveform Chart** și **Waveform Graph**.

4. Se va crea un instrument virtual care să genereze un semnal sinusoidal similar celui descris în 4.3, bazat pe modulul **Sine Wave**, care să aibă următoarele caracteristici: amplitudinea 10, frecvența semnalului 100Hz, faza inițială 120 de grade, iar frecvența de eșantionare se va alege astfel încât să rezulte câte 10 eșantioane pe o perioadă a semnalului. Apoi, se va modifica frecvența de eșantionare astfel ca să se obțină câte 50 și 100 de eșantioane pe perioadă și se vor comenta efectele rezultate. Gradarea axei **X** se va face astfel ca în toate cazurile să se vizualizeze câte cinci perioade.

5. Se va crea un instrument virtual, asemănător celui descris în 4.3, pentru generarea și afișarea simultană a două semnale, unul de tipul **Square Wave** și celălalt de tipul **Sawtooth Wave**.

6. Utilizând procedura descrisă în 4.4.1, se va realiza un instrument virtual pentru trasarea graficului funcției trigonometrice inverse **arcsin(x)** folosind modulul dedicat care se află în fereastra **Trigonometric** (fig.4.26).

7. Se va trasa graficul unei elipse prin metoda nodurilor, cu formula descrisă în 4.4.2. Se va ține seama că ecuația elipsei raportată la axele ei de simetrie este

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

Pentru semiaxele elipsei se vor lua valorile $a=3$ și $b=2$.

4.6. Modul de lucru și prezentarea rezultatelor

Pentru lucrările de la punctul 1, directorul **Aplicatia II.4** se caută în **My Documents**. Experimentările se vor efectua în concordanță cu relatările cuprinse în secțiunile respective din lucrare.

La punctele 2 și 3, pentru a distinge cât mai bine cele două grafice, se vor modifica: culorile, tipul și grosimea liniilor, stilul punctelor, etichetele scalelor, formatul și precizia de reprezentare a valorilor pe cele două scale. Se va afișa lista de cursoare, se va defini un cursor și apoi se vor schimba caracteristicile acestuia.

Referitor la punctul 4, se va modifica frecvența de eșantionare astfel ca să se obțină câte 50 și 100 de eșantioane pe perioadă și se vor observa efectele rezultate. Gradarea axei **X** se va face astfel ca în toate cazurile să se vizualizeze câte cinci perioade.

La punctul 5, se vor varia caracteristicile celor două semnale și se vor preciza corelațiile necesare pentru vizualizarea corespunzătoare a acestora.

Cu privire la punctul 6, pentru gradarea axelor se va ține seama că abscisele vor conține valori ale funcției sinus, iar ordonatele valori ale arcelor corespunzătoare.

Pentru punctul 7, se vor observa corelațiile necesare între valorile mărimilor de intrare, modul de gradare a axelor și numărul de iterații. Se vor urmări cu atenție indicațiile din **4.4.2**.

Fiecare student, în fișierul grupei sale, urmărind punctele de la chestiunile de studiat, va înscrie răspunsurile, rezultatele și comentariile proprii. Pentru **VI**-urile descrise în lucrare și cuprinse în biblioteca de programe sunt suficiente rezultatele din experimentări cu comentariile aferente. Pentru **VI**-urile nou create, prevăzute în tematicile specificate la Chestiuni de studiat, se vor prezenta și panourile frontale și diagramele bloc respective, însoțite de imagini cu rezultate, eventuale explicații și comentarii.